

# ATARI COMPUTER ENTHUSIASTS

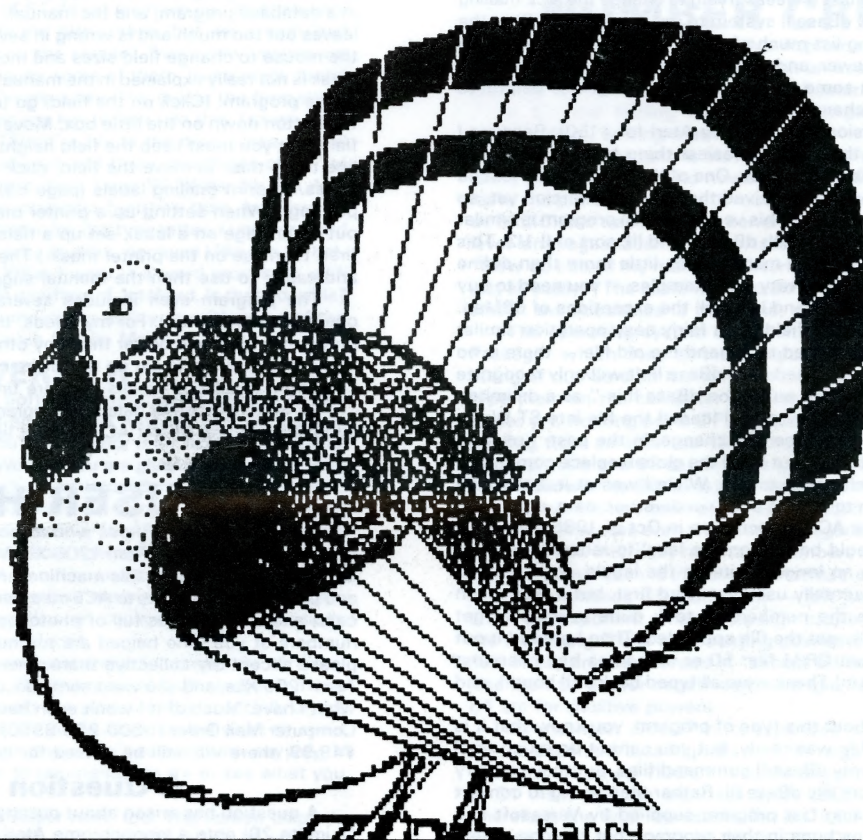
3662 Vine Maple Dr. Eugene OR 97405

---

**NOVEMBER, 1986**

Mike Dunn, Jim Bumpas, Larry Gold, co-editors

---





# News and Reviews

Mike Dunn, Co-Editor

The two biggest items this month are the beginning of a series of articles on **TurboBASIC** and the Portland regional Atari show. TurboBASIC is a European public domain BASIC for the 8-bit XE/XL Atari. There are many new features, including new commands, full Atari BASIC compatibility, great increase in speed, and an excellent compiler for even greater speed. Brought to you with complete documentation and sample programs on a double-sided disk put together by the Western N.Y. Atari User Group. Other user groups may obtain a licence and disk for distribution for \$15 from them (POB 59, Buffalo, NY 14216). For ACE members, only \$10 — every XE/XL owner who uses or programs in BASIC really should get one!

The Portland Atari Show was the third regional show sponsored by Atari, and was great fun. The most impressive things to me were the **Magic SAC**, David Small's Mac cartridge. It really works (I ordered one and will report more when I receive it), Atari's **Write** which is similar to Microsoft Word and very impressive, and the new hard disk/3½" floppy by **Astra**. This unit was very small, quiet and cool, and if I ever get the money, it will be the one I might want (assuming it works well). Astra also has a very nice, inexpensive and pretty mousepad, which really does make the difference — \$7 retail, in blue (Astra, 2500 S. Fairview, Santa Ana, CA 92709).

Other very impressive items included Paul Heckel and his wonderful new **Zoomracks II**. Paul acted as a private tutor to anyone who wanted help and was again delightful to talk to. Many of the vendors had unbelievably low prices on their products, as many have been reduced and had special show prices. The \$495 Graphic Artist is now \$195, and Easy Draw was \$80. Atari showed new 8-bit software as well as the blitter chip.

One of the items I bought at the show was a monitor cable allowing me to hook a composite color monitor to my monochrome system. In low res, such as a neochrome picture or program, it is as good as the Atari color monitor, and at med res, it is very readable and much better than I expected. Using a monochrome system for word processing and a color composite for drawing programs and games is an excellent cost-effective choice, especially if you already have the monitor from your 8-bit system.

I spent a good part of the last 6 weeks trying to change the ACE mailing list from my ATR 8000 CP/M dBase II system to my new 520ST. Over the years, I have kept the mailing list much of the time, and as we've grown we've had to change to newer and bigger databases. In my reviews following, I will also explain some techniques which might be helpful to those of you who need to change databases.

**dBMAN** (Versasoft version, now sold by Atari for \$150). Because I had been using dBASE II, I thought the easiest thing to do might be to use the dBase II/III "compatible" database. One of the local dealers loaned me a copy to try out; they had not received the new Atari version yet, so I used the store copy of version 2.0. This very powerful program is similar to, but frustratingly different from both dBase II and III; sort of II 1/2. This program is not GEM based, and the manual does little more than define the commands and functions with very few examples — you need to buy a book on dBase III (not dBase II) and learn all the exceptions of dBMan.

The first step is to set up the fields — a fairly easy operation similar to what I am used to. Then I needed to append the old file — there is no way to specify the delimiter you need as in dBase II. It will only recognize "," in the text mode. The standard output on dBase II is ";" as a delimiter. Rather than go back to the CP/M system, I loaded the file into ST Writer and changed with the global replace and change. In the past, I used to write programs to do such a thing, but I find the global replace commands in word processors are much easier to use. While I was at it, I changed the entire date code system to a more easily understood date, e.g., in the old format if you began your ACE subscription in Oct of 1985 your code would be 10P5 and you would be told on the label to renew in Sept of 86. With bulk mail, we can no longer write on the labels, so I changed it to Sep 86 on the label. I actually used 1st word first, but it was much slower and couldn't handle the numbers. It took quite a while to get everything correct, but finally got the file appended. Then I printed it out and compared it with the old CP/M file; 50 or so names had been lost somewhere by being loaded in! These were all typed back in (I hope!), and the file was ready.

If you know anything about this type of program, you know that it is more like a language. The file was ready, but you cannot do much until you write a program. I tried my dBase II command files, but they are very different since dBMan is more like dBase III. Rather than trying to convert them, I took the demo Mailing List program supplied by Versasoft and changed it. There were many bugs in their program, but the conversion was not too difficult. The main problem is I was unable to make command files in the program — you need to use a word processor, and keep switching back and forth during debugging. This phase took about 3 weeks of spare time, but finally, all was working, but unbelievably slow in some operations. As an example, one of the files took about 5 minutes to index on the CP/M system took 20 minutes with dBMan! Thinking either there was something wrong with this version or perhaps the new Atari version was better, I called the company to find out. The person who I spoke to refused to compare versions, said it was against company policy to compare their product with dBase II, and that the new version was "presumably" much better but would

not tell me how as this was also against company policy. Well, the program works, is very powerful, requires lots of effort and programming ability and seems bug free. It is different enough from Ashton-Tate's products to be frustrating if you use programs from books, etc., and is very slow. Since my needs were simple, I returned it to the dealer and was going to go back to the old system, when a review copy of the following arrived by UPS.

**DataRieve** (Abacus, POB 7219, Grand Rapids, MI 49510, \$50). Everything about this program is about the opposite of the above. It is an inexpensive, GEM based data management program. It is not a programming language, and not a powerful, difficult to use relational database. It is very quick and easy to use, with many surprising features.

First, I exported the file from dBMan, and again, could not specify the delimiter, so it used ",". With DataRieve, you can specify any delimiter for separating the fields and records, but only 2 characters for each ("," is three characters). Back to ST Writer to remove the ",". Setting up the fields is extremely easy — you don't need to even specify the width, as it is a free form database. The first time, I made some mistakes, and found a bug — I couldn't get the delete file command to work, so I made a new file. Setting up the fields is so quick and easy, there was no problem. Then I indexed the file — I thought it would be a long wait, but in less than 1 minute it was finished! Then to find and modify or add new records — pure joy! Every other database or mailing list program I had used in the past required you to load in a separate command file or program for either adding or editing existing records. Since I don't always know if a check from you is a renewal or a new membership (you might be amazed how many checks I get with no information about what it is for — please write on your check what you want), it is necessary for me to go back and forth frequently, with some programs taking a few minutes to load in the correct module. In DataRieve, you put your mouse on the gearshift lever and immediately change from search mode to edit mode. You can do most anything with the mouse; but you also have the option of using function keys or control codes for most anything — your choice. The program uses screen and printer masks to do what you want, and you can make these masks as simple or elegant as you wish. Other outstanding features include many search options, ability to index on up to 20 fields, cut and paste options, ability to very easily create new fields, change field sizes, make subranges and subfiles, and import and export various types of files. Things I don't like about the program is the copy protection, which I feel is inappropriate in a database program, and the manual. The manual is clearly written but leaves out too much and is wrong in several places. As an example, using the mouse to change field sizes and moving around the screen or printer mask is not really explained in the manual and is very important in operation of the program. (Click on the field, go to the right lower corner and hold the button down on the little box. Move horizontally to change size of the field, but you must keep the field height full to work — stay on the lower line to do this. To move the field, click within the field, not on the little boxes. To print mailing labels (page 69), use 6 rather than 999 for lines per page. When setting up a printer mask, do the screen mask first. To put a message on a label, set up a field of 0 length on the screen mask first, then use on the printer mask.) The program is much more powerful and easier to use than the manual suggests.

The program even includes several size ramdisks for even faster operation (not tested). For my needs, the speed and ease of use of this program is ideal and better than any other I have used for the mailing list in the last 6 years. To set up the entire mailing list took only a hour or so, compared to several weeks of spare time for dBMan.

As we receive other database programs to review, I will review as above by loading in one of the ACE mailing list files.

## USER HINTS

I want to thank all those who responded to my request for help with documentation for the Atari 1200 XL. My son, my nephew and my niece are all proud owners of this machine, and now they can use it more fully and productively, thanks to ACE members. I received long-distance phone calls, and the mail was full of photocopies and original documents. The numbers of you who helped are too numerous to thank individually, so please accept my collective thanks here. By the way, if any of you have Atari 1200 XLs, and you want some documentation, write me and I'll share what I have. Much of it I won't even have to photocopy again. As long as Computer Mail Order (1-800-233-8950) continues selling the 1200 XL for \$49.99, there will still be a need for documentation support.

## A Question of Color

A question has arisen about putting a color printer (one, say, like the Okimate 20) onto a monochrome Atari ST. Will graphs and other output be printed in different colors for the different textures of the graph (or other graphic)? Or will the output be monochrome only? Anyone who has tried this combination, please let the readers here know about it. Thanks. — JB

## STuff

Our ST Group meetings are every 4th Wednesday of each month (we've been meeting regularly since December, 1985) at 7.00 pm. The location is Amazon Community Center, in the south crafts room.



## BUMPAS REVIEWS

I went to Portland Sunday the 12th to see the Atari Expo sponsored by the Portland Atari Club and Atari Corp. Ticket sales are rumored to have been around 2,000. There were more than 100 Atari computers (both 16 and 8 bit) running a wide variety of software. There were several new items I had not seen before. One vendor had a touch-screen modification to the ST computer. The Mill Camp, a restaurant in the Eugene-Springfield area spent \$5000 for 5 of them. Customers point to menu items on a NeoChrome screen. The selection is immediately displayed to the kitchen where the food is prepared and at the waitress station where the bill is prepared.

I was also curious about the Magic Sac. It was fun seeing Macintosh software running madly on a couple of STs. I also saw demonstrated several new programs I had not known existed for the ST. The blitter chip was there, happily moving parrots across the screen. Previous animations without the blitter had only shown a single parrot moving on the screen. There was music (MIDI synthesized) and examples of nearly everything available in hardware and software for the ST, and a lot for the 8-bits, too.

I spent most of my time at the QuickView Systems booth talking to Paul Heckel about his new product, **Zoomracks II**. The new product looks very good, and appears to be a significant improvement over the older Zoomracks, which is a good product itself. The screen display now gives you more system information, with rack names and help information more clearly displayed. Math functions are included. Macros and special output formats (labels, letters, etc.) seem to be more easily created. We'll have a more complete review of this product by next issue, I expect. I had some particular questions, because I have a project to convert about 500k of data from a Macintosh for a friend who's switching to the Atari ST. His data is mail-list data, which is easily handled by Zoomracks. The toughest part of the project for me was to clean up all the spaces and Macintosh formatting from the data file. I just put it in ST Writer, set up the global search and replace and let it run. ST Writer is S-L-O-W! I think the program took more than 24 hours to go through about a 300k segment of this data to make all the replacements I wanted. At least I didn't have to sit at the console — I just checked it every few hours or so. I do know it was 24 hours, within an hour or so, because it finished just before dinner one day and I was impatient and checked it twice in an hour period.

By the way, the latest version of ST Writer we have our librarian calls version 1.5 (it says version 1.07 on the title screen, but we already had a version 1.07 without this notice on the screen). Every new version we hear "the bugs are dead!". Well, maybe some of them. I've already found some bugs in this latest version: Create a new file, delete the formatting line at the top, and press the Backspace key. When I do this, I crash back to the desktop. Also, everytime I "Print" a file to the disk, I lose several bytes off the end of the file. So I always add three short lines of "x" at the end so I don't lose any important data. I use ST Writer for the ACE newsletter, and I find it has more powerful features than 1st-Word — especially for preparing this newsletter for the typesetter. I use 1st-Word for my correspondence and documents at work, because I like the "what-you-see-is-what-you-get" screen. What I want is a word processor with the power of ST Writer and the GEM interface of 1st-Word. Is that possible? Maybe also with graphics and column-sizing ability as with some of the desktop publishing packages available for the Mac.

## UNIVERSE II

**Universe II** (\$80, Omnitrend Software) might seem familiar to you who've played this game on the 8-bits. But be warned: This ST game is much more of a game than what you saw on your 8-bit Atari. In this game, you are an agent of the Federated Worlds. You are given assignments from time to time. All the while, you may disguise yourself as a commerce trader, an ore miner, or a pirate. You will receive money for some of your assignments, but you must also make money from your occupation. With more money you can buy bigger and better spaceships, shuttles, weapons and other accoutrements. You must also hire and pay the wages for your crew. You can trade at the markets in various star systems. You can engage in planetary and atmospheric mining. You can conduct combat operations both in deep space and on planet surfaces.

The familiar starport markets and spaceship functions are all there, hyperspace and sublight travel, etc. But the mouse! It all seems so much easier with all the menu selections on the screen and just a click of the button. Even easier, many times all the menu choices you want to avoid are "dimmed out" with light print and you can't select them anyway. It sure saves a lot of time going back to the manual to try to see what you want.

Trading at the market is much easier now, too. The screen displays more information than we're used to. Now it tells us the class of item (i.e., weapon, lifeform, medicine, etc.). This way it's much easier to avoid taking illegal products to your destination. You no longer have to memorize the product class for products such as "Endico Khive". The messages displayed on the screen seem longer and more complete.

There are a couple of new modules added to this game not available on the 8-bit version. While the whole game is really a text adventure (the graphics consist of GEM menus and simple solar and star maps, and a game grid), a traditional text parser is added for the Starport where you can interact with other characters. And a game grid is added for the assault module. This is used to move your marines in a tactical engagement with

enemies at planetary mining sites or in spaceships you might assault.

There are 3 disks in the game, and with a 1-meg system you can put up a ramdisk to cut down the disk I/O and speed up the game. Disk A is write protected, but the files can be copied to a ramdisk. The manual says Disk A must be in the default drive. But it doesn't seem to do any disk reads during the game. Maybe you can take it out after the program is booted up. If you have a double-sided drive, disks B and C can be put on a double-sided disk. This way you have no disk swaps and all your disk A I/O can proceed at ramdisk speed.

One thing worries me about this game, especially if you don't have enough memory for a ramdisk. The program seems to write player files to each disk. This isn't so bad with disks B and C. These are not write protected, so you can work from copies. But disk A is write protected, and the average user won't be able to create a back-up disk for his own use easily. With a ramdisk, the player file destined for disk A goes on the ramdisk. When you quit, you can copy this file to a separate data disk and avoid writing to disk A. Just remember to copy this player file back to the ramdisk the next time you play. So, if you enjoy a giant game of interstellar commerce, mining and piracy, this game will give you months of enjoyment.

## WRITE 90

**Write 90** (Xlent Software, \$30) is a handy little utility program which prints ascii files from a disk to a printer sideways. Your spreadsheets which usually will be wider than the paper in your printer can now print down the page, and even across several pages. You have a choice between 5 font sizes from 12 characters per inch to 4.5 characters per inch. There are drivers on the disk to support Epson, Citizen, Panasonic and Star Micronics printers. Also included on the disk is a disk formatting program permitting over 400k of data on a single-sided disk and over 800k on a double-sided disk. I'm using this utility instead of the desktop formatter to get the extra data space. This package does everything it promises, which is rare in an age of advertizing hype. Xlent Software is doing an excellent job of providing owners of the Atari ST with utility software.

— Jim Bumpas

## NUMBER CRUNCHER

By A.J. Aspromatis

½krench-er ½ vb 1 : that which an assembly language programmer must be when he wants to use mathematics.

n 1 : the feeling in the pit of a programmer's stomach as he contemplates writing math routines.

That it will happen is inevitable. There will come a time in the life of every programmer, be he novice, hacker, or professional, when he must leave the prefabricated math routines of the higher level languages (such as BASIC) and tread over the barren lands of integer arithmetic in the lower level languages (such as C or Assembler). The question will then come: "How do I do all my math?!" Fortunately, the answer following will come just as easily: "The same way you did them before!" As it so happens (what else?), the substance is the same; only the form will change.

The operating system of the Atari has a "Floating Point Arithmetic Package" which does all the computer's math operations. It can take a string of numbers (in the way one might normally write them by hand or type them on the computer keyboard) and change them into integer or, better yet, floating point (FP) representation. In this latter form, numbers ranging from as small as 10 to the negative 98th power ("1" trailing a decimal point and 96 zeros) to as great as 10 to the positive 98th power ("10" followed by 98 zeros) can be represented. Thereafter, they can be added, subtracted, multiplied, and divided with 9 or 10 decimal units precision. There are even provisions for logarithm and polynomial evaluation!

The Atari internally maintains numbers as binary code decimal (BCD) floating point. Each is in a six byte format: one byte of exponent and five bytes of mantissa (10 BCD digits — left justified). The most significant bit of the exponent reflects the sign of the number (plus or minus) with the remaining 7 bits representing powers of 100 (rather than powers of 10). These are represented in "excess 64 notation", meaning that values less than 64 are for negative powers and values equal to or greater than 64 are for positive powers.

For example: -901.27 in its expanded form is  $-9.0127 \times 100 \pm 1$ . So for the exponent byte high bit = 1 for negative number & remaining bits = 65 for +1 power ( $100 \pm 1$ ) so the exponent byte is 1100 0001 (binary) or C1 (hex) and the mantissa bytes (BCD) are 09 01 27 00 00 (hex). Therefore, the final representation is C1 09 01 27 00 00 and the special case of "0.0" is "00 00 00 00 00". Of course, the programmer need not concern himself with this internal structure if all he wants to do is take strings of digits, operate on them, and display the results. The only problem remaining then is accessing these routines the same way those higher level languages do. Thanks to the tight, structured programming of the Atari designers, it is no problem at all. Actual FP operations are performed by setting reserved memory locations, which hold incoming values, and calling the proper fixed-address ROM routines. "Cruncher" takes most of the effort out of using the more important operations.



## Text Tidier 1.0

by Dave Yearke

As you may recall, we received from a Dutch users' group Turbo BASIC, a language for XL/XE computers which is compatible with Atari BASIC but offers over 60 new features and is three times faster. Well, they've done it again, this time they sent us the Turbo BASIC Compiler, a program which compiles BASIC programs so they run three times faster than Turbo BASIC! I wanted to write a program to demonstrate the power of these two new programs, so after a few nights of work Text Tidier was born.

Text Tidier is designed to eliminate much of the editing going into processing files downloaded from such services as CompuServe. These files often have excess spaces for justification, carriage returns every line, and all kinds of other annoying things which must be weeded out. The most bothersome types of text files are those which use the Control-J Control-M combination for linefeed and carriage return. For MS-DOS machines this is fine, but our Atari's use a totally different character.

In its simplest form, Tidier strips out linefeeds ( $\pm J$ ) and turns ASCII carriage returns ( $\pm M$ ) into ATASCII returns (code 155). It also breaks down the file into smaller pieces to fit into a word processor like AtariWriter or PaperClip. I've downloaded files over 100K long, far too big to fit into any word processor for the Atari! When you run the program, the first thing it asks for is the maximum file size for the output files. 12,000 is pretty good for AtariWriter, although this depends on what DOS is being used, whether a printer driver is loaded, and so on. For PaperClip, I've found on my XL about 24,000 bytes can be loaded, although I recommend about 20,000 so you have room to edit the file. Next it asks for the Input file. Put the disk in the drive and enter the name. If you don't enter a device like "D:" it will default to drive one. All Output files will be named "SPLIT.0xx", where "xx" is the number of the file being written, starting at zero. Be careful not to overwrite any existing files with the same name!

Now come the formatting parameters. When it asks "CONVERT RETURN TO SPACE (Y/N)" type "Y" if you want all carriage returns to be converted. This is nice if you want to reformat the text in your word processor. This option only converts single returns; if it encounters more after the first one it will leave them alone because it assumes that it is a break between two paragraphs. The next option, "TIDY UP SPACING (Y/N)", is probably the most powerful. It will get rid of multiple spaces between words, excess spaces before return characters, convert ASCII Tab characters ( $\pm I$ ) to 5 spaces, and will make sure two spaces follow all periods, colons, question marks, and exclamation points, and one space follows all commas, even if it has to add them. One note: it will not add spaces to periods which have no spaces after them, because it can't tell if it is part of a decimal number. The next option, "PROMPT BEFORE WRITING (Y/N)", should get a "Y" response if you need to switch disks between reading and writing, and an "N" response if you want to let it go without interruption. The "BREAK FILES ON SPACE (Y/N)" simply assures you a word won't be split over two files (this may not work if the last word before writing is over 15 characters long, but that's pretty rare). The last option is a convenience to those with the AtariWriter or PaperClip word processors. If you select either "A" or "P" at the prompt, it will write the "chain next file" command at the end of all files except the last one. In addition, it will put a paragraph symbol where it thinks it's appropriate, and convert the ASCII formfeed character ( $\pm L$ ) to its AW or PC equivalent. If you select "N" for "neither", then the paragraphs will be indented five spaces and formfeeds will be left alone.

By the way, the program only allows character codes within the range 26-127, plus CR (13) and FF(12). Characters higher than 127 have the high bit stripped to make them "normal," while control characters are ignored. If you want to allow inverse and control characters (although this can be a potential problem with true ASCII files; control characters can have bizarre meanings, and true ASCII does not allow character codes higher than 127. If you get any, it's probably a transmission error), change line 215 to: 215 REPEAT :GET #1,N:UNTIL (N does not = LF) This will only filter linefeed characters.

This program was written out of necessity, but I also had a lot of fun doing it. If you have any questions, comments, or suggestions, please let me know (I can be found lurking at the meetings or on BATES in the "Ask Dave Y!" message base). I want to correct any bugs you find (gasp!) quickly.

## ST Library

Our local club council discussed making a change in the ST Library. I've been the librarian for over a year now. Ralph Walden has done such a good job with the ACE BBS — he's really got it so it's setting "world-class" standards now. He's put together probably one of the best ST sections in the world. We found that he and I were really duplicating a lot of work. Actually, I was getting most of the best stuff for the library from him. As the sysop of the ACE BBS, he gets all the "latest and greatest" public domain things. So Ralph Walden is the new ST Librarian. Send all your library business to him.

## THE TURBO COMPILER

Documentation and Operating Instructions

by Dave Arlington Original Program by Frank Ostrowski

This documentation is provided by the Western New York Atari Users Group and may be reprinted freely provided this credit is included.

I know everyone is still rubbing their hands in glee over TURBO-BASIC, the new public domain Basic interpreter imported from Holland and Germany. This month we are proud to present a new treat from our international friends overseas, the TURBO COMPILER. The TURBO COMPILER will increase the speed of your TURBO BASIC programs 3-5 times and regular Atari BASIC programs can be speeded up to 10-15 times faster. Unfortunately, like TURBO BASIC, the TURBO COMPILER will only run on the XL/XE series of computers. In addition, the finished compiled programs will also only run on the XL/XE series.

The TURBO COMPILER is very easy to operate. It can be used with more than one disk drive, even Drive 8, the 130XE ramdisk. On the TURBO COMPILER disk, you will find several files. The important ones for the operation of the TURBO COMPILER are COMPILER.COM and RUNTIME.COM. Before you begin, you should prepare two disks. The first one should be a blank formatted disk with DOS.SYS and DUP.SYS written on it. Your second disk should contain your BASIC program (TURBO or ATARI BASIC).

To begin, insert the TURBO COMPILER disk into your drive and turn your computer on. When it has finished loading you will be in TURBO BASIC. Type DOS to get to the DOS 2.5 menu. Choose DOS Option 'L', Load Binary File and load the file COMPILER.COM. After a short while, you will see a screen full of German. If you do not read German, do not worry, that is what this article is for! At any time you are on this screen, you may reboot your system by hitting Control-R or you may return to DOS by hitting Control-D. You will be asked if you really want to do this. If you do, don't forget to 'J' for Ja and not 'Y' for Yes!!

To compile your programs, remove the TURBO COMPILER disk from the drive and insert the disk with the program you want to compile. (If you have two drives, insert your disk with the program to compile in Drive 2 and the blank DOS disk in Drive 1.) Press the number 1 key (Number 2 if you have two drives). A complete listing of all the files on that drive will appear on the screen with one highlighted in inverse video. Using the arrow keys, highlight the file you want to compile and hit Return. That's all there is to it!

At the top of the screen, you will see the line numbers fly by as the program is compiled. When it is finished, you will be prompted for a file name to save your compiled program under. It has to have an extender of CTB (for Compiled Turbo Basic). The program will not let you use any other extender. At this time, if you have one drive, you should remove your disk with your original program on it and replace it with your blank DOS disk. If you want your compiled program to be an AUTORUN file, you should name it AUTORUN.CTB. Then the file will automatically load and run at bootup time.

There is one more step you must do to get a completely runnable compiled program. Go back to DOS and copy the file RUNTIME.COM from the TURBO COMPILER disk to the disk with your compiled program on it. Rename the RUNTIME.COM file to the name AUTORUN.SYS. Your disk is now ready to go. The TURBO COMPILER does not compile to runnable code, you must run the RUNTIME.COM file to run your compiled program.

One technical note on the TURBO COMPILER: If you're not familiar with other compilers for the ATARI computers, you should know many have trouble compiling certain types of statements or require you to organize your program structure in a certain way. So far, none of these problems have occurred on any programs we have tested with the TURBO COMPILER. The only statement we've been unable to compile so far is the END statement. The only program structure we have had trouble with is a FOR-NEXT loop with two or more NEXTs for one FOR. For example: 10 FOR X=1 TO 50 IF X=3 THEN NEXT X 30 NEXT X

Other than those two examples, we have had no problems compiling any type of program, either TURBO or ATARI BASIC. Please let us know of any problems you run into compiling any types of statements or programs.

Whether you program in TURBO BASIC or just want to turbocharge your old Atari BASIC programs to super speed, I'm sure you will find the TURBO COMPILER will fit all needs.

## INDEX(STR1,STR2)

This is a C function implemented in assembly. There is some controversy about how the index() function in C should be defined. In this month's assembly program I have written the function to search for the occurrence of one string in another. The function will return the address in string 1 that string 2 appears, or a zero if the string was not found. In your C header file, you should define index() as a char pointer.

— Ralph Walden



# WORD PROCESSOR

(by David Castell XLEnt Software \$29.95)

XLEnt Software has added a word processor to their lineup of programs. They have an extensive printware series and were only missing a word processor to make it complete. I am surprised it took them this long to do one since they seem committed to print programs and the like.

This word processor is different from most in that it uses a joy stick to access many of its functions and it has a print spooler. While it will run on any Atari with 48k or more some of its functions will not run except on the 130XE.

When you boot up this program down in the lower right hand corner of the screen six boxes appear in a set of 3 on top of 3 more. Each of the boxes has an icon in it. Glasses for the search function, a disk for disk utilities, camera for copy, scissors for cutting, paste bottle for repositioning the cut material, and a large P for the printing commands. With the joy stick you can move an arrow to the icon you want and when you press the fire button you are put into that mode. The escape key gets you back to the edit mode to go on with your work. There are all the usual commands for backspace, chaining files, margins, headers, footers and everything else you want. One of the things I do not like about this program is the use of 3 keys to get the word processor to do what you want; i.e., SHIFT+CONTROL something to do some task. I much prefer only 2 key controls; this goes for any program I use.

The one function I find to be of the most use is the spooler. With this I can send material to the printer and do not have sit and wait till it is printed out so I can go on with my work as the printer is printing. This function only works with the 130XE. If you do a lot of word processing this may be the program for you and it is priced low enough you can buy it and just use it when you only need certain of its features.

The use of the joy stick to move about to access the icons smacks of a poor attempt at trying to have a fake mouse. This is one feature I can take or leave as I have enough to do without having to reach for the joy stick to get to a new function, although some of you might find it nice.

Overall I find this to be a good word processor with features found on the "high priced spreads". It offers alot and comes with something which really puts the icing on the cake, good documentation which is clear, concise, and readable, now how is that for a program. Keep it up XLEnt and I will keep my praises coming.

— Larry Gold

# DOTS-PERFECT

Do you ever wish your old trusty Epson printer had the neat near letter quality modes and the push button print mode selection the new FX85+ series has? For less than \$80, you can purchase an upgrade kit for your Epson MX, RX, FX or JX printers and add the latest "plus" functions to your printer. The **Dots-Perfect** (Dresselhaus Computer Products, 837 East Alostia Ave., Glendora, CA 91740 818-914-5831) upgrade kit enhances your printer with features like NLQ (near letter quality) mode, push button selection of 160 font styles, the IBM PC graphics character set, and a buffer reset command.

I ordered a kit for my 3 year old MX80 from Alpha Scientific, Box 626, Chesterfield, MO 63017 (314-532-5282) at a total cost of \$72.73. The kit appeared on my doorstep the day after the order was phoned in. The upgrade kit consists of a small circuit board which replaces 3 EPROM chips in the printer. No soldering is required. All you need for installation is about 10 minutes and a flat blade and phillips screwdrivers. The instructions are very clear and include many photographic aids. Dresselhaus provides a phone number in the event assistance is needed. The excellent documentation also includes numerous examples on how to test the upgrade and invoke the new fonts and functions via the Epson panel switches or software.

The front panel selection means you can program the print mode of the Epson by pressing various combinations of the "On Line", "FF" and "LF" buttons. For example, pressing the FF button results in the printer giving you a short beep and places the printer in NLQ mode. This is fantastic for those programs which don't let you set up the printer. Pressing the LF button puts you back in draft mode and gives you a long beep to confirm the selection. Other printer functions selectable from the buttons include condensed, double wide, emphasized, double strike, perf. skip, 1/2" left margin, italics, underline, fine print, 8 lines/in., slash zero, and 8-1/2" paper. An adhesive label summarizing the operation of Dots-Perfect can be attached to the front of the printer.

The only criticism I have is I wish the Epson MX80 also included the pica character set provided with the RX and FX series. The "morse code" scheme of selecting print modes can become confusing, but this is also true of the Epson Plus series.

Dresselhaus gives a 1-year warranty. They also claim the proper installation of the upgrade will not void the Epson printer warranty.

— Ron Robinson

# OASIS BBS

When I took over work on the OASIS BBS program (then called PD/M) one of my main goals was to make it work well for both 8 and 16 bit computers. A lot of clubs have BBS's running on 8 bit computers, and have found to their dismay, that the BBS works poorly, if at all, with an ST caller. Most of the 8 bit BBS's will not accept uploads or downloads to an ST. ST BBS's will work with an 8 bit computer, but you're stuck with an 80 column menu on a 40 column screen. The OASIS BBS program has overcome this problem so well that many of our ST callers assume we are running on an ST.

OASIS asks the user what their screen width is, and then uses word wrap on all text for the user's screen width. You can optionally setup 3 different menus, one for 8 bit Atari callers, one for 80 column callers, and one for 40 column ASCII callers. You can have up to 5 distinct download areas so you can put 8 bit files in one area, and ST files in another. OASIS supports Xmodem, CRC, Ymodem, and text (with optional word wrap) downloads, and Xmodem and CRC uploads. OASIS is written in assembly — it can easily keep up with any ST terminal program even with a 9600 baud direct connect to an ST. If you have an 8 bit BBS and you want to support ALL Atari users, this is definitely a BBS program worth looking into. If you want to see an OASIS running, call ACE at (503) 343-4352 Sunday-Friday, or HELP (316) 683-7514

— Ralph Walden.

# BBS NEWS

With Express 3.0 adding CRC, and the OASIS BBS software adding CRC and Ymodem, there have been a lot of questions raised about the difference between the various download protocols. For a long time, Xmodem has been the standard way to transfer a file between your terminal program and a BBS. Xmodem sends 128 bytes of data at a time, plus 4 bytes of information your terminal program uses to see if the data came through correctly. If there was noise on the phone line, your terminal program will usually recognize it and ask the BBS to send the block of 128 bytes again. USUALLY this means an error free file, but Xmodem is not perfect, and occasionally errors will slip through, and your file will be bad. The program may run fine right up until it gets to the garbled section, and then crash. CRC is a more accurate method of transferring files. It sends 5 bytes of control information for every 128 bytes, and uses a sophisticated mathematic algorithm to check for errors. It is almost impossible to get a bad file with a CRC download. Ymodem is an extension of CRC. Instead of sending 128 bytes at a time, it sends 1,024 bytes at a time. This will make for a slightly faster file transfer for a direct call, and will halve the download time if using a long distance service like PC Pursuit. On the other hand, if you have a bad phone connection, Ymodem will be slower since each bad transfer will cause 1,024 bytes to be resent instead of 128 bytes.

If your terminal program supports it, you should always use either CRC or Ymodem. ST users should set their terminals to default to CRC. If your terminal supports Ymodem, it will switch automatically to Ymodem if you tell the BBS that you want Ymodem protocol. 8 bit users, if you don't have a copy of Express 3.0, you should get one — you can usually find it on a BBS. So far, only the 850 version is up to 3.0. Rumor has it that the 1050 and MPP versions will be held up until the author completes version 3.1 which will support Ymodem.

— Ralph Walden, SYSOP.

# VP RAMBLINGS

So far, the response to our survey is not as great as we hoped, but it does give us an indication of where you want the newsletter to go and what you want in it. We will try to comply with your wishes and see if we can do what you want. Since we made the newsletter bigger and we hope better please get your renewals in as we can't go on without your support and this comes from renewing your subscription. You can find out when you are to renew by looking on the cover of the newsletter and checking the date on the label.

We had an Atari expo in Portland and it seems to have gone over pretty well. Atari had a booth manned by Atari executives. Along with them there were 3rd party vendors and various other people showing their wares. It was quite good for a first time event in the Northwest. If it becomes an annual event it will be because of the Portland Atari Club which put it on. My hat goes off to you for the fine job you did.

With Christmas just a short time away we should start seeing new products for all the machines so we can stuff the Christmas stockings with Atari goodies for the kids and of course ourselves. Now that Atari is going public we can now find out just how well (we hope) they are doing and maybe even put some pressure on the company to bring out some of the things we have been asking for. Let's hope so.

— Larry Gold



## ANALOG ST

(Reprint: by Lucas Lozo and Ron Cork from KAOS Magazine)

A lot of us bought an ST for two reasons: To get the latest in micros at a very good price, and to get a computer capable of producing good quality, high-resolution graphics. We therefore got the monochrome version. Later we began to realize that without a color monitor, we were missing out on some of the good fun software written only for the color system. If you have only recently purchased an Atari ST and are lucky enough to get the built-in composite video output and RF modulator then you're one step in the right direction for those great color games. If you have only a mono system, then your only choice is to beg, borrow or buy a color monitor at considerable extra cost.

Atari, in their infinite wisdom, have followed the yellow brick road laid down by Apple with the IIc and used non-standard connectors for the floppy drives, DMA port and monitor port. Apple even used a different type of connector from the previous II series, supposedly to force the gullible public to buy only Apple, or in our case, Atari add-ons as well as give us all bloody big headaches and munch merrily on our bank books.

The big trick is to find an analog color monitor. Most color monitors sold around the traps are digital, meant to be driven by those Big Blue beasts and so, in their unmodified condition, are not suitable at all. But they are a damned sight cheaper than an analog one and work fine, if you can convince someone to convert it to analog. It is not a very expensive or difficult operation, but it does require a fairly specialized knowledge and circuit diagrams of the monitor.

I (Lucas) originally purchased a mono ST and planned to use the analog RGB monitor I bought for my Apple. But, of course, the connector was absolutely useless. The ST monitor connector is a 13 pin DIN (DIN meaning standard, but just try to find one), so the only choice left was to make my own. I started with some standard 0.1" spaced PCB IDC header pins which come in strips of about 20 pins. I cut 3 strips of 4 pins each and superglued these into 3 rows. A single pin was glued smack dead-center on the bottom, 4th row (check out the diagram), which gave me a 12 pin matrix with a tail. In similar fashion I made a female socket to allow the mono monitor to plug into my adapter. I already had a connector for my color monitor, and along with a phono socket for the external amplifier, all were mounted in a small box for rigidity with a short length of trailing cable with the home-made connector going to the ST socket. Another way to make the connector is to use a piece of veroboard and cover it in epoxy cement. George used this method on his Mitsubishi monitor. I (Ron) used the original connector, a 15 pin d socket and a short length of lead. Marc Adams also used this method. The 13 pin DIN comes apart very easily.

The Atari ST has separate vertical and horizontal sync signals coming out of the computer. Most monitors use a composite sync signal (mine did), so I overcame this little obstacle by using two diodes to gate the signals together. You could use an AND gate here instead but I didn't have one on hand at the time. The diodes work fine. One of the signals from the Atari is the Monochrome Detect, which tells the ST which type of monitor is plugged in. A simple switch in this line will allow both monitors to remain plugged into the adapter box with just a flick of the switch to change between them. Unfortunately, the ST seems to keep a watchful eye on this line and will still re-boot when a switch change is made.

The audio sound line also comes out of the monitor connector and it is a simple matter to connect a phono socket on these. You can use either a simple type or one which will disable the monitor amplifier/speaker when you plug in your own external amp or hi-fi (try the bouncing ball with a lot a big fat sound — great).

As you will see on the diagram, I have put 1k trim pots in the RGB lines to allow for fine tuning of the colors. Apart from the ground pin, all others are left unconnected. With both monitors turned on, the one not being used will have a black screen. My monitor is a Kaga Vision I RGB with both analog and digital inputs. The one you choose must have at least an analog input. If you already have a digital RGB, then this must be converted over before it will be of any use. The usual trick is to include a small amplifier/driver circuit between the digital input and the RGB gun drivers. It is up to you who you can to do the job. Have fun.

## SWAP MEET

Our December meeting will be the semi-annual swap meet and WISTEC benefit. If you want to participate by selling or trading any of your computer hardware or software items, contact Larry Gold. Give him a \$5 donation for WISTEC and you can set up your things on a table. WISTEC will give receipts to those requesting it.

## TURBO BASIC IS HERE NOW

## LITTLE PEOPLE COMPUTER

There is a little person living in your computer. And you give him a home. He comes in and looks around. After he looks around he goes and gets his bag and his dog. When he comes back with his bag he unpacks his bag. After he moves in you have to make sure he has food and water. To give him water you have to push ctrl."W". To make sure he has food you push ctrl."F".

If you want to know his name at first you ask him to type you a letter. When he types you a letter he will type the date and who it is to. After he types a letter he will sign his name.

When you want to give him a phone call you push ctrl."C". When the phone rings he sits in his chair and you can pet him. To pet him you have to push ctrl."P". When you pet him a hand comes out of the wall and it pets the side of his head. The LCP loves to play games. The games he likes to play are Anagrams, Card War, 5-Card Draw Poker, Black Jack, and Word Puzzle.

The house that your LCP lives in is three story. At the top there is a piano. If you ask him please he will play it. There is a record player up there too. If you ask please he will play that too. You can give him a record by pushing ctrl."R".

When he goes to bed you should not let him sleep too long. He would rather have your attention. To get him out of bed you have to set the alarm off. To do that you have to push ctrl."A".

If the LCP has run out of food and water he will turn green. This means he is sick. When he is sick he will just lie on the bed. To get him well you have to give him food and water.

One other thing the LCP loves to do is watch TV. He watches TV for a long time.

The LCP only goes to bed when he wants to. Before he goes to bed he takes a bath, put on his jammies, and brushes his teeth.

We think it is a fun game. We think other kids will like it too.

—Laurie and Lani Cook

## TURBO BASIC

(from WNY Atari Users Group)

Turbo Basic Disk Help File (This is what is included in the TurboBasic 2-sided disk for \$10 from ACE. ed.)

Hi! Welcome to possibly the greatest public domain program of 1986! If you have not tried Turbo Basic yet, try this simple test. Boot the old Atari Basic with one of your favorite Basic programs, preferably a nice SLOOOOW one. Now boot the other side of this disk and run the same program. Pretty impressive, huh? Now that you have an inkling of what you've got in your hot little mitts, let me tell you what is included on this disk.

SIDE 1: TURBO BASIC - the AUTORUN.SYS file The TURBO COMPILER - COMPILER.COM RUNTIME.COM - The runtime package for Turbo compiled programs. TIDIER.TUR - A great program for anyone who uses downloaded text files (or the text files on this disk!!) MAGIC.TUR - A Koala type drawing program written in Turbo Basic. PMMOVE.TUR and PUTGET.TUR - Two demo programs showing off some of TB's power.

SIDE 2: README - Hopefully obvious by now! TIDIER.TXT - Documentation for using Text Tidier TURBODOC.TXT - A complete translation of every Turbo Basic command, feature, and function. (PLEASE NOTE: Unlike the other text files on this side of the disk, this file is in PaperClip format. When printed out with the PaperClip Word Processor, the result is a nicely formatted three page command list). TCOMPILR.TXT - Complete instructions on using the TURBO COMPILER.

Please enjoy using this disk. We are always looking to publish new Turbo programs in our newsletter, read by over 500 people. If you want to see your program published in POKEY, the newsletter of the W.N.Y. Atari Users Group, please submit them (along with any comments you may have on this disk) to:

W.N.Y. Atari Users Group  
P.O. Box 59

## STuff

Our ST Group meetings are every 4th Wednesday of each month (we've been meeting regularly since December, 1985) at 7.00 pm. The location is Amazon Community Center, in the south crafts room.

Our public domain library now also contains a disk from Stone Age Software, Inc., Box 1216, Amherst, NH 03031. It's a "partialware" disk with two programs: **LEWIS123** and **ENCRYPT**. Stone Age explains "partialware" software has 90% of the "functionality" of the full program, which can be purchased for \$30.

**LEWIS123** is a chemist's spreadsheet which might be valuable to chemistry students. It also contains the ability to draw interactive models of molecules to help illustrate bonding. The drawing files can be saved in DEGAS or MEGAFIL formats.

New additions to the library include POOL.PRg (monochrome only), a game; and SPOOLER33K.PRg and SPOOLER.ACC.



BLITZ  
by A.S. Dewdney

This game is a version of one of the classic computer games and will run on any Atari.

You are the pilot of an old bomber flying low over a deserted city. Suddenly one of your engines fails and you begin to rapidly lose height. The situation looks hopeless when you suddenly realise that you have a full bomb bay! Your only chance is to bomb all the buildings flat so that you can land safely. If you land safely you are awarded 1000 points before going on to another city. The buildings will be higher and your bombs will become less effective. To make things a little easier you have on board 3 emergency boosters which will provide some extra height. You will be awarded an extra booster after clearing each city.

The program is written using Graphics 0 which only permits 2 colours. To make the game more interesting players have been used to form the sun, clouds and an extra plane.

The effects used in the title page are achieved using a Colourflow program. When you crash, the whole screen appears to shudder which is done scrolling the screen up and down. The game also awards ratings and includes a hi-score feature.

Happy Landings.

```
1 REM *****
*
2 REM *          BLITZ
*
3 REM *          BY A.S.DENDNEY
*
4 REM * -----
*
5 REM * PAGE 6 MAGASINE - ENGLAND
*
6 REM *****
*
7 REM
9 RESTORE 30000:GOSUB 24000
10 GRAPHICS 0:POKE 752,1:POSITION 14
```

```
,10:?"PLEASE WAIT"
11 SETCOLOR 2,9,8:SETCOLOR 1,0,0:SET
COLOR 4,3,8
12 GOSUB 20000:FOR T=53420 TO 53251:
POKE TT,0:NEXT T
13 GOSUB 9000:?"K"
15 GOSUB 7000:SC=0:B5=4:PP=3
16 ? "K":POKE 82,2:P5P=SP*2
17 POKE 53248,160:POKE 53249,70:POKE
53250,110
20 Y=21:?"K":S=0
21 COLOR 93:PLOT 0,1:DRAWTO 39,1:PLO
T 39,1
25 POSITION 4,0:?"<<BLITZ>>" B_"
;PP
26 H$=STR$(HISC):POSITION 26-(LEN(H$
)-1),0:?"HISCORE_";HISC
27 PX=250:POKE 53251,PX
28 POSITION 15,22:?"&"SC;
30 FOR T=4 TO 32 STEP 2
40 FOR G=0 TO HI+(RND(0)*8)-4
50 Y=Y-1:POSITION T,Y:?"K"
60 NEXT G:POSITION T,Y:?"4"
70 Y=21:NEXT T
80 FOR T=5 TO 33 STEP 2
90 FOR G=0 TO HI+(RND(0)*6)-3
100 Y=Y-1:POSITION T,Y:?"K"
110 NEXT G:POSITION T,Y:?"4"
120 Y=21:NEXT T
125 SOUND 0,255,12,6
130 X=1:Y=3:TRAP 300
135 REM MAIN LOOP
140 POSITION X,Y:?" ,./"
141 IF STRIG(0)=0 AND BOMB=0 AND Y<1
9 THEN GOSUB 1000
142 IF BOMB=1 THEN GOSUB 1020
143 IF X=29 AND Y=19 THEN GOTO 6000
145 IF S=1 THEN SC=SC+10:POSITION 21
,22:?"SC;
148 PX=PX-P5P:POKE 53251,PX
150 X=X+5P:IF X=35 THEN X=0:Y=Y+1:P
OSITION 35,Y-1:?" ";
160 LOCATE X+4,Y,0:IF D<>32 AND D<>3
5 THEN GOTO 10000
170 IF STICK(0)=14 AND Y>3 AND PP>0
THEN Y=Y-1:POSITION X,Y+1:?" ":S
OUND 1,100,14,15:GOSUB 500
180 GOTO 140
300 TRAP 300:PX=220:GOTO 140
500 PP=PP-1:POSITION 28,0:?"PP
999 REM BOMB MOVEMENT
1000 BOMB=1:BCH=0
1010 BX=X+1:BY=Y+1:RETURN
1020 IF BY=18 THEN POSITION BX,BY:?"
":BOMB=0:BCH=0:SOUND 1,0,0,0:5=0:
RETURN
1021 BY=BY+1
1023 IF S=1 THEN SOUND 1,30,0,9:5=0
1025 POSITION BX,BY:?"K"
```

```
1026 POSITION BX,BY-1:?" "
1030 LOCATE BX,BY+1,BD
1040 IF BD<>32 THEN BCH=BCH+1:5=1:IF
BCH>85 THEN BCH=0:BOMB=0:POSITION B
X,BY:?" ":5=0:SOUND 1,0,0,0:RETURN
1060 RETURN
5999 REM WELL DONE!
6000 SOUND 0,0,0,0:SOUND 1,0,0,0:BOM
B=0:5=0:BCH=0:CITY=CITY+1
6005 IF CITY=4 THEN GOSUB 13000:CITY
=1
6010 POSITION 12,10:?"CONGRATULATIO
NS!"
6015 IF HI=11 AND B5>4 THEN B5=B5-1
6016 IF HI>11 AND B5<5 THEN SP=-1
6020 FOR T=1 TO 40:SOUND 1,205-(T*2)
,8,15:SC=SC+25:POSITION 21,22:?"SC;
SOUND 1,0,0,0:NEXT T
6030 FOR T=1 TO 150:NEXT T:IF HI<12
THEN HI=HI+2
6040 POSITION 11,11:?" GET READY
!"
6050 FOR T=1 TO 100:NEXT T
6060 GOTO 16
6999 REM PLAYERS
7000 P=PEEK(106)-8
7005 RESTORE 7070
7010 POKE 54279,P:PMB=P*256:POKE 559
,46
7020 POKE 53277,3
7030 POKE 704,30
7050 FOR T=PMB+512+35 TO PMB+512+42
7060 READ A:POKE T,A:NEXT T
7070 DATA 60,126,255,255,255,255,126
,60
7080 POKE 705,14
7090 POKE 53257,3
7100 FOR T=PMB+640+50 TO PMB+640+57
7110 READ A:POKE T,A:NEXT T
7120 DATA 60,126,255,255,255,255,126
,60
7130 FOR T=PMB+768+42 TO PMB+768+47
7140 READ A:POKE T,A:NEXT T
7150 DATA 16,56,124,255,62,24
7155 POKE 706,14:POKE 53258,3
7160 PX=250:POKE 53251,PX
7170 POKE 707,62:POKE 623,9
7180 RESTORE 7200:FOR T=896+PMB+45 T
O 896+PMB+52
7190 READ A:POKE T,A:NEXT T
7200 DATA 12,25,51,254,48,24,12,4
7210 RETURN
7999 REM RATINGS
8000 FOR T=53248 TO 53251
8010 POKE T,0:NEXT T
8020 SOUND 1,0,0,0:SOUND 0,0,0,0:SET
COLOR 2,9,8
8030 ? "K"
8035 IF SP=1 THEN SC=SC+1000
```



# CRUNCHER BY ASPROMATIS

8-BIT

```

0100 ;SAVE #D:CRUNCHER.ASM
0110 ;.....

0120 ;Number crunching routines
0130 ; using Atari's floating
0140 ; point package.
0150 ;Examples shown in "C" style.
0160 ;
0170 ; Start date: 2/23/85
0180 ; Rev. date: 9/25/86
0190 ;
0200 ; by Arthur J. Aspromatis
0210 ;.....

0220 ;
0230 ;Atari floating point package
0240 ; variables & ROM routines.
0250 ;
0260 FR0 = $D4
0270 FR1 = $E0
0280 FLPTR = $FC
0290 INBUFF = $F3
0300 CIX = $F2
0310 LBUFF = $0580
0320 ;
0330 APF = $D800
0340 FASC = $D8E6
0350 IFP = $D9AA
0360 FPI = $D9D2
0370 FADD = $DA66
0380 FSUB = $DA60
0390 FMUL = $DADB
0400 FDIV = $DB28
0410 ZFR0 = $DA44
0420 ZF1 = $DA46
0430 FLD0P = $DD8D
0440 FLD1P = $DD9C
0450 FST0P = $DDAB
0460 ;
0470 ;.....

0480 ;TXTFL
0490 ; Routine to get ASCII text and

0500 ; change it to floating point
0510 ; format.
0520 ;[V0]=TXTFL([V1],[V2])
0530 ; V1 -pointer to text buffer
0540 ; V2 -pointer to f.p. buffer
0550 ; V0 -returned error code
0560 ;.....

0570 ;
0580 .IF .REF TXTFL
0590 TXTFL
0600 ;Get pointer to ASCII text
0610 LDY #0

0620 STY CIX ;Offset to 1st
ASCII byte
0630 LDA V1
0640 STA INBUFF ;ASCII buf.poi
nter
0650 LDA V1+1
0660 STA INBUFF+1
0670 ;Floating Point Arithmetic Pack
age converts to f.p.
0680 JSR APF
0690 BCC TX0
0700 LDA #$FF ;INVALID ASCII
code
0710 JMP ECODE
0720 ;Transfer f.p. number to string

0730 TX0
0740 LDA V2 ;Get f.p. buffe
r pointer
0750 STA FLPTR
0760 LDA V2+1
0770 STA FLPTR+1
0780 JSR FST0P
0790 LDA #0 ;OPERATION COMP
LETE code
0800 JMP ECODE
0810 .ENDIF
0820 ;
0830 ;.....

0840 ;FLTXT
0850 ; Routine to get floating point

0860 ; and change it to ASCII text
0870 ; format.
0880 ;[V0]=FLTXT([V1],[V2])
0890 ; V1 -pointer to f.p. buffer
0900 ; V2 -pointer to text buffer
0910 ; V0 -returned error code
0920 ;.....

0930 ;
0940 .IF .REF FLTXT
0950 FLTXT
0960 ;Get pointer to f.p.
0970 LDA V1
0980 STA FLPTR ;pointer to f.p
.
0990 LDA V1+1
1000 STA FLPTR+1
1010 ;Load floating point # to buffe
r
1020 JSR FLD0P
1030 ;F.P. package does the conversi
on
1040 JSR FASC
1050 ;Transfer ASCII number to strin

1060 LDA V2 ;Setup pointer
1070 STA FR0
1080 LDA V2+1
1090 STA FR0+1
1100 LDY #0
1110 FT1
1120 LDA (INBUFF),Y ;Get ASCII
text
1130 BMI FT0 ;If inverse, la
st one
1140 STA (FR0),Y ;Put it to us
er buffer
1150 INY
1160 CPY #99 ;Max.buf.size
1170 BCC FT1
1180 BCS FT2 ;BRA
1190 ;---
1200 FT0
1210 AND #$7F ;Inverse "end-o
f-number" flag to norm.
1220 STA (FR0),Y
1230 INY
1240 FT2
1250 LDA #0 ;Add NULL
1260 STA (FR0),Y
1270 LDA #0 ;OPERATION COMP
LETE code
1280 JMP ECODE
1290 .ENDIF
1300 ;
1310 ;.....

1320 ;MATH
1330 ; Routine to ADD,SUBTRACT,
1340 ; MULTIPLY,DIVIDE floating poin
t
1350 ; numbers.
1360 ;[V0]=MATH([V1],[V2],[V3],[V4])

1370 ; V1 -pointer to f.p. buffer
1380 ; V2 -math operation character
1390 ; V3 -pointer to f.p. buffer
1400 ; V4 -pointer to answer buffer
1410 ; V0 -returned error code
1420 ;.....

1430 ;
1440 .IF .REF MATH
1450 MATH
1460 ;Get f.p. numbers to operate on
1470 LDA V1 ;1st f.p. buffe
r
1480 STA FLPTR
1490 LDA V1+1
1500 STA FLPTR+1

```



1510 JSR FLD0P ;Get f.p. number	2010 ;	ment
1520 LDA V3 ;2nd f.p. buffer	2020 ;.....	2490 EOR #\$FF
1530 STA FLPTR	2030 ;FLINT	2500 CLC
1540 LDA V3+1	2040 ; Routine to get floating point	2510 ADC #1
1550 STA FLPTR+1	2050 ; and change it to signed integer	2520 STA (FLPTR),Y
1560 JSR FLD1P ;Get f.p. number	2060 ; format.	2530 INY
1570 ;Get math operator	2070 ;[V0]=FLINT([V1],[V2])	2540 LDA FR0+1
1580 LDA V2	2080 ; V1 -pointer to f.p. buffer	2550 EOR #\$FF
1590 CMP #'+	2090 ; V2 -pointer to integer var.	2560 ADC #0
1600 BEQ MA1	2100 ; V0 -returned error code	2570 STA (FLPTR),Y
1610 CMP #'-	2110 ;.....	2580 LDA #0 ;OPERATION COMP
1620 BEQ MA2		LETE code
1630 CMP #'*	2120 ;	2590 JMP ECODE
1640 BEQ MA3	2130 .IF .REF FLINT	2600 ;---
1650 CMP #'/'	2140 FLINT	2610 FL3
1660 BEQ MA4	2150 ;Get pointer to f.p. buffer	2620 LDA FR0 ;Put integer
1670 LDA #\$FE ;INVALID OPERATION code	2160 LDA V1	2630 STA (FLPTR),Y ; to variable.
1680 JMP ECODE	2170 STA FLPTR ;Load up FR0	2640 INY
1690 ;Perform math operation	2180 LDA V1+1	2650 LDA FR0+1
1700 ;---	2190 STA FLPTR+1	2660 BMI FL2 ;Overflow if high-bit set
1710 MA1	2200 JSR FLD0P	2670 STA (FLPTR),Y
1720 JSR FADD	2210 ;Allow negative integers	2680 LDA #0 ;OPERATION COMP
1730 BCC MA5	2220 LDA FR0	LETE code
1740 MA1A	2230 BPL FL1A	2690 JMP ECODE
1750 LDA #\$FD ;RESULT OUT OF RANGE code	2240 EOR #\$80 ;Toggle negative off	2700 .ENDIF
1760 JMP ECODE	2250 STA FR0	2710 ;
1770 ;---	2260 LDA #1 ;Flag negative	2720 ;.....
1780 MA2	2270 BNE FL1B ;BRA	2730 ;INTFL
1790 JSR FSUB	2280 ;---	2740 ; Routine to get signed integer
1800 BCC MA5	2290 FL1A	2750 ; & change it to floating point
1810 BCS MA1A	2300 LDA #0 ;Else flag positive	2760 ; format.
1820 ;---	2310 FL1B	2770 ;[V0]=INTFL([V1],[V2])
1830 MA3	2320 STA LBUFF	2780 ; V1 -pointer to integer var.
1840 JSR FMUL	2330 ;Convert to integer	2790 ; V2 -pointer to f.p. buffer
1850 BCC MA5	2340 JSR FPI	2800 ; V0 -returned error code
1860 BCS MA1A	2350 BCC FL0	2810 ;.....
1870 ;---	2360 FL2	
1880 MA4	2370 LDA #\$FC ;F.P. IS NEGATIVE OR >=65535.5 code	2820 ;
1890 JSR FDIV	2380 JMP ECODE	2830 .IF .REF INTFL
1900 BCS MA1A	2390 ;Transfer integer to variable	2840 INTFL
1910 ;Put result into answer buffer	2400 FL0	2850 ;Get pointer to integer
1920 MA5	2410 LDA V2 ;Get variable location	2860 LDA V1
1930 LDA V4 ;Get answer buffer pointer	2420 STA FLPTR ;Use F.P. pointer	2870 STA FLPTR ;Load up FR0
1940 STA FLPTR	2430 LDA V2+1	2880 LDA V1+1
1950 LDA V4+1	2440 STA FLPTR+1	2890 STA FLPTR+1
1960 STA FLPTR+1	2450 LDY #0	2900 LDY #1
1970 JSR FST0P	2460 LDX LBUFF ;Flag neg.?	2910 LDA (FLPTR),Y ;Negative?
1980 LDA #0 ;OPERATION COMP	2470 BEQ FL3	2920 BPL IN0
LETE code	2480 LDA FR0 ;Yes-2's complement	2930 DEY ;Yes-uncomplement
1990 JMP ECODE		nt
2000 .ENDIF		2940 LDA (FLPTR),Y
		2950 SEC



# CRUNCHER CON'T

```

2960 SBC #1
2970 EOR #$FF
2980 STA FR0
2990 INY
3000 LDA (FLPTR),Y
3010 SBC #0
3020 EOR #$FF
3030 STA FR0+1
3040 LDX #1 ;Flag neg.
3050 JMP IN1
3060 ;---
3070 IN0
3080 DEY
3090 LDA (FLPTR),Y ;Use F.P. p
ointer
3100 STA FR0
3110 INY
3120 LDA (FLPTR),Y
3130 STA FR0+1
3140 LDX #0 ;Flag pos.
3150 IN1
3160 STX LBUFF
3170 ;Convert to F.P.
3180 JSR IFP
3190 ;Adjust sign
3200 LDA LBUFF ;Neg.flagged?
3210 BEQ IN2
3220 LDA FR0 ;Yes-toggle sig
n
3230 EOR #$80
3240 STA FR0
3250 IN2
3260 ;Transfer F.P. to string
r pointer
3270 LDA V2 ;Get F.P. buffe
r pointer
3280 STA FLPTR
3290 LDA V2+1
3300 STA FLPTR+1
3310 JSR FST0P ;Transfer
3320 LDA #0 ;OPERATION COMP
LETE code
3330 JMP ECODE
3340 .ENDIF
3350 ;
3360 ;.....

3370 ;Return error code
3380 ;.....

3390 ECODE
3400 STA V0
3410 RTS
3420 ;
3430 ;-----
3440 ;User variables
3450 V0 .BYTE 0
3460 V1 .WORD 0

```

```

3470 V2 .WORD 0
3480 V3 .WORD 0
3490 V4 .WORD 0
3500 ;
3510 ;-----*

```

WNY Atari Users Group  
Turbo Basic Disk  
Help File

This is what is included in the TurboBasic 2-sided disk for \$10 from ACE..ed.

Hi! Welcome to possibly the greatest public domain program of 1986! If you have not tried TurboBasic yet, try this simple test. Boot the old Atari Basic with one of your favorite Basic programs, preferably a nice SLOCCOW one. Now boot the other side of this disk and run the same program. Pretty impressive, huh? Now that you have an inkling of what you've got in your hot little mitts, let me tell you what is included on this disk SIDE 1:

TURBO BASIC - the AUTORUN.SYS file The TURBO COMPILER - COMPILER.COM RUNTIME.COM - The runtime package for Turbo compiled programs.

TIDDER.TUR - A great program for anyone who uses downloaded text files (or the text files on this disk!!)

MAGIC.TUR - A Koala type drawing program written in Turbo Basic.

PMOVE.TUR and PUTGET.TUR - Two demo programs that show off some of TBs power.

SIDE 2:

README - Hopefully obvious by now!

TIDDER.TXT - Documentation for using Text Tidier

TURBODOC.TXT - A complete translation of every Turbo Basic command, feature, and function. (PLEASE NOTE: Unlike the other text files on this side of the disk, this file is in PaperClip format. When printed out with the PaperClip Word Processor, the result is a nicely formatted three page command list.

TDCOMPILR.TXT - Complete instructions on using the TURBO COMPILER.

Please enjoy using this disk. We are always looking to publish new Turbo programs in our newsletter, read by over 500 people. If you would like to see your program published in POKEY, the newsletter of the W.N.Y. Atari Users Group, please submit them (along with any comments you may have on this disk to:

W.N.Y. Atari Users Group  
P.O. Box 59  
Buffalo, N.Y. 14216

# TURBO BASIC IS HERE

10

# WALDEN'S ASM

```

.globl _index
16-BIT

.text

* index(str1,str2)
* return address of str2 in str1
* return 0 if not found

* get address of string to search
_index: move.l 4(sp),a0
* address of string to search for
move.l 8(sp),a1
* get a character from the string
index: move.b (a0)+,d0
* if end of string, return a zero
beq.s false
* match?
cmp.b (a1),d0
* no, keep looking
bne.s index
* save current addresses
movem.l a0-a1,-(sp)
* backup to matched character
subq.l #1,a0
* first char matches, check rest
indexlp: move.b (a0)+,d0
* end of string
beq.s maybe
* get search character
move.b (a1)+,d1
* end of string, we have a match
beq.s match
* are they the same
cmp.b d0,d1
* yes, keep looking
beq.s indexlp
* restore the original addresses
nomatch: movem.l (sp)+,a0-a1
* go look for another match
bra.s index
* are we at end of search string
maybe: tst.b (a1)
* yes, we have a match
beq.s match
* restore stack
addq.l #8,sp
* return a zero
false: moveq #0,d0
rts
* restore the original addresses
match: movem.l (sp)+,d0/a1
* backup to matched character
subq.l #1,d0
* address of match in d0
rts
*****

```



**TURBO BASIC COMMAND LIST**  
Compiled and Translated by  
Dave and Laura Yearke

This documentation is provided by the Western New York Atari Users Group and may be reprinted freely provided this credit is included.

In case you've just landed from Mars, or just plain haven't heard yet, TURBO BASIC is the exciting new Public Domain Basic Interpreter that we recieved from the Atari Users Group in Holland. It works on the XL or XE series of Atari computers. It's almost too good to be true and should be a definite must for all XE or XL Atari owners. Turbo BASIC, in addition to offering 42 more commands and 22 more functions than Atari BASIC, gives the user 1603 more bytes of program space by "hiding" part of itself under the XL/XE's operating system. It also runs 3 times faster than Atari BASIC, includes most DOS commands, has advanced graphics and programming functions, and is insensitive to lower case or inverse characters for most commands.

Name	Syntax	Description
Name	Syntax	Description

**TURBO BASIC COMMANDS:**

**Disk I/O**

BLOAD	BLOAD "D:name"	Binary loads file name (DOS option L with /N).
BRUN	BRUN "D:name"	Binary load and run file name (DOS option L).
DELETE	DELETE "D:name"	Deletes the file name (DOS option D).
DIR	DIR	Disk directory (DOS option A).
	DIR "Dn:*.*"	Directory of drive n, note that wildcard extenders may be used.
LOCK	LOCK "D:name"	Locks the file name (DOS option F).
RENAME	RENAME "D:old,new"	Renames the file name (DOS option E).
UNLOCK	UNLOCK "D:name"	Unlocks the file name (DOS option G).

**Graphics**

CIRCLE	CIRCLE x,y,r	Plots a circle with center at x,y and radius r.
	CIRCLE x,y,r,r2	R2 is an optional "vertical radius" for true circles or ellipses.
CLS	CLS	Clears the screen.
	CLS #6	Clear screen opened in channel

6.

FOOLOR	FOOLOR n	Determines fill color.
FILLTO	FILLTO x,y	A fill command analagous to the BASIC commands "POSITION x,y: XIO 18,#6,0,0,"S:"
PAINT	PAINT x,y	Another type of fill command, this one is a recursive routine that will fill any closed object as long as x,y are inside it.
TEXT	TEXT x,y,a\$	bit-blocks text in a\$ at x,y.

**Memory**

DPOKE	DPOKE m,v	Pokes location m,m+1 with 2-byte integer v (0 <= v <= 65535).
MOVE	MOVE m,m1,m2	Block transfer; moves m2 (number of bytes) from starting position m to new starting position m1.
-MOVE	-MOVE m,m1,m2	Same as MOVE but copies starting with the last byte of the block.
BPOT	BPOT #n,adr,len	Block Put; same as FOR I=0 TO len-1:PUT #n,PEEK (adr+I):NEXT I
BGET	BGET #n,adr,len	Block Get; same as FOR I=0 TO len-1:GET #N,A: POKE adr+I):NEXT I
ZPUT	ZPUT #n,a	Until now, there was no convenient way to put numeric values onto disk or cassette files other than by using PRINT, which converted them to strings first, a slow and cumbersome process. ZPUT puts the

number to the device "as is," in 6-byte FP format.

ZGET	ZGET #n,A	Get a number stored with ZPUT from the device and store it in variable A. Again, this is much faster than using "INPUT #n, A".
------	-----------	--

**Structured Programming**

REPEAT	REPEAT	Start a REPEAT-UNTIL loop.
UNTIL	UNTIL <C>	Terminate when condition <C> met.
WHILE	WHILE <C>	Start a WHILE-WEND loop to end when condition <C> met.
WEND	WEND	Terminate a WHILE-END loop.
ELSE	ELSE	Optional extension for IF. The IF condition must not be followed by a "THEN", but terminated by end-of-line or colon.
ENDIF	ENDIF	Ends an IF-ELSE-ENDIF or IF-ELSE condition. Note that this allows an IF condition to span more than one BASIC line, provided the "IF" statement is structured as shown in Note 4.
DO	DO	Starts an "infinite" DO loop.
LOOP	LOOP	Cycle back to the start of a DO loop.
EXIT	EXIT	Exit a DO-LOOP loop.
PROC	PROC name	Start definition of procedure.
ENDPROC	ENDPROC	End definition of procedure.
EXEC	EXEC name	Execute procedure name.

**General Programming**

PAUSE	PAUSE n	Pause processing for n/50 seconds.
RENUM	RENUM n,i,j	Renumber the program starting at line n, first number is i, increment is j. This function will handle GOTOs, TRAPs, and all other line references except those which involve variables or computed values.
DEL	DEL n,i	Delete lines n-i.
DUMP	DUMP	Display all variables and values. For numeric arrays, the numbers are the DIMed values plus one. For strings, the first number is the current LENGTH of it and the second number is the DIMed size of it. DUMP also lists procedure names and labels with their line values.
	DUMP name	DUMP to device name, such as "P:" or "D:DUMP.DAT".
TRACE	TRACE	Trace program during execution.
	TRACE -	Turns trace mode off (Default).
DSOUND	DSOUND n,f,d,v	Form of SOUND which activates channel-pairing for increased frequency range.
	DSOUND	Turns off all sounds.
GO TO	GO TO n	Alternate form of GOTO.
*L	*L	Turn line-indent on (Default).
	*L -	Turns line-indent off.
*F	*F (or *F +)	Special mode for FOR..NEXT loops which corrects a bug in Atari BASIC. Seems that in Atari BASIC, an "illegal" reverse loop like "FOR X=2 TO 1:PRINT X:NEXT X" will execute once even though the condition is met initially (X is already greater than 1). Turbo BASIC fixes this bug, but leaves it available for Atari BASIC programs which may take advantage of it.
	*F -	Turns off the special FOR..NEXT mode to make Turbo BASIC act like Atari BASIC.
*B	*B (or *B +)	Command which allows the break key to be trapped via the "TRAP" command within a program.
	*B -	Turns off the special BREAK key mode.
-	-	Special form of REM which puts 30 dashes in a program listing.

**Line Labels**

#	# name	Assigns the current line number to the label name. This is a convenient way to get around the problem of renumbering when using variables as line numbers. Labels can be thought of as a special form of variable, as they occupy the variable name table along with the "regular" variables. We also believe that the number of variables allowed has been increased from 128 to 256 to allow for the addition of these labels.
GO#	GO# name	Analogous to the GOTO command.



## Modifications

**CLOSE** **CLOSE** Close channels 1-7.  
**DIM** **DIM a(n)** Will automatically assign a value of zero to all elements of the numeric array being dimensioned, and null characters to all elements of a string (The LEN is still variable, however, and initially zero).  
**GET** **GET name** Wait for a key press, assign the value to name. Same as "OPEN #7,4,0,"K":GET #7,name:CLOSE #7".  
**INPUT** **INPUT "text";a,b...** Prints text as a prompt before asking for variable(s), same as Microsoft-BASIC.  
**LIST** **LIST n,** List program from line n to end.  
**ON** **ON a EXEC n1,n2,...** Variation of ON...GOSUB for procedures. N1, n2 and so on are names of procedures to be run.  
**ON** **ON a GO# n1,n2,...** Similar to ON...GOTO except that line labels are used instead of line numbers.  
**POP** **POP** This command now pops the runtime stack for all four types of loops.  
**PUT** **PUT n** Same as "PRINT CHR\$(n)";  
**RESTORE** **RESTORE #name** Restores the data line indicated by the label name.  
**RND** **RND** Parentheses are no longer needed at the end of this command, but it will still work if they are there.  
**SOUND** **SOUND** Turn off all sounds.  
**TRAP** **TRAP #name** TRAPs to the line referenced by the label name.

## TURBO BASIC FUNCTIONS:

### Arithmetic/Logic

**HEX\$** **HEX\$(n)** Convert n to hex string.  
**DEC** **DEC(a\$)** Convert hex string A\$ to decimal.  
**DIV** **n DIV i** Integer quotient of n/i.  
**MOD** **n MOD i** Integer remainder of n/i.  
**FRAC** **FRAC(a)** Fractional part of a.  
**TRUNC** **TRUNC(a)** Truncates fractional part of a.  
**RAND** **RAND(n)** Generates random number 0-n.  
**\$** **\$nnnn** Allows input of hexadecimal numbers, but they are converted to decimal. Ex: "FOR I=\$0600 to \$067F" => "FOR I=1536 to 1663".  
**&** **n & i** 8-bit boolean AND.  
**!** **n ! i** 8-bit boolean OR.  
**EXOR** **n EXOR i** 8-bit Exclusive-OR.

### Memory

**DPEEK** **DPEEK(m)** Double-PEEK of m,m+1.  
**TIME** **TIME** Time of day(numeric).  
**TIME\$** **TIME\$** Time of day string, HHMMSS.  
 Unfortunately, the time commands don't work properly because they were written for European Ataris which operate at 50 Hz, instead of 60 Hz like American ones, the net result being that they gain 12 minutes each hour.  
**INKEY\$** **INKEY\$** Returns last character typed.  
**INSTR** **INSTR(x\$,a\$)** Returns relative location of start of string A\$ within X\$ (returns 0 if not found). The match must be exact; strings with the same letters but differences in case or type (normal or inverse) will not be found.  
**INSTR(x\$,a\$,i)** i specifies the starting point of the search.  
**UINSTR** **UINSTR(x\$,a\$)** Same as INSTR, does not distinguish between case or inverse characters. Ex: UINSTR("Hello","hello") returns 1.  
**UINSTR(x\$,a\$,i)** Specifies optional starting point.  
**ERR** **ERR** Value of last error number.  
**ERL** **ERL** Line last error occurred at.

## Constants

%0  
 %1  
 %2  
 %3

These four constants simply stand for the numbers 0-3, respectively. The difference with using these in a program is that "%=1" requires 10 bytes, whereas "%=1" only needs 4 (numbers require 7 bytes, 6 for the number plus an identifier preceeding it. It is always a good practice to make variables for numbers that are used more than three times in a program).

## NOTES:

1. Variable, Procedure and Label names may contain the underscore ( ) character.
2. To print a double-quote (") in a text string, use two of them together, instead of the Atari BASIC method of using CHR\$(34). Ex: "TEST";CHR\$(34);"TEXT" becomes "TEST""TEXT" in Turbo-BASIC, both of which produce the output => TEST"TEXT.
3. Upon initial boot-up, TURBO-BASIC looks for a BASIC file named AUTORUN.BAS. If it finds an AUTORUN.BAS file, it will automatically load and run this file.
4. Turbo-BASIC also prints out English descriptions of all errors, including several new ones for errors involving the new commands:

Error - 22 ?NEST = Loops not properly nested.  
 Error - 23 ?WHILE = WEND with no corresponding WHILE.  
 Error - 24 ?REPEAT = UNTIL with no corresponding REPEAT.  
 Error - 25 ?DO = LOOP with no corresponding DO.  
 Error - 26 ?EXIT = EXIT is outside a loop.  
 Error - 27 ?XPROC = Error executing PROC.  
 Error - 28 ?EXEC = ENDPROC with no corresponding EXEC.  
 Error - 29 ?PROC = Procedure does not exist.  
 Error - 30 ?# = Label does not exist.  
 Also, Error 15 has been expanded to include an UNTIL which relates to a REPEAT which has been deleted.  
 4. A multiline IF is constructed like this:

```
10 IF X > 10
20 PRINT X-10
30 GO# TOO_BIG
40 ELSE
50 PRINT X
60 GO# X_IS_OK
70 ENDIF
```

Note also the use of line labels in the GOTO statements.

—Dave & Laura Yearke

## AVATEX 1200

The Avatex 1200 is the first 1200 bps modem selling for less than \$100. Avatex advertises the modem as being 99% Hayes compatible. These claims might be more accurate if they stated the modem is compatible with the commands **used** 99% of the time. It has no internal speaker, does not support the Hayes "S" register set and requires the baud rate to be selected by a switch from a push button on the front panel.

I personally really miss the internal speaker function. You can get around this by keeping a phone nearby and listening in after the modem is finished dialing. The lack of the "S" register means the user can't program such things as how many times the phone rings before the modem answers the call. This should only affect those planning to use the modem to run a BBS. The baud rate switch is a minor annoyance in normal use which is easily fixed by using separate 300 bps and 1200 bps phone lists. The modem does automatically detect and set baud rate when answering a call. It does seem the modem should have been capable of automatically setting baud rate when originating calls also.

The Avatex works fine with 850 Express, Modem 7, Homepak, VT100, Chameleon, Flash!, and ST Talk software terminal programs. It also operates successfully with the Atari 850 Interface and the ICD P:R: Connection.

Front panel indicators include: DTR, SD, RD, HS, MC, TM and RI, plus option switch settings for 8/7 bits, loopback/self test modes, and DTR-CTS-CD disable. The Avatex Hayes AT Command References include: AT - Attention; P - Pulse Dial; , - Pause 4 sec; AA - Auto Answer; A - Answer; Z - Modem Reset; D - Dial; T - Touchtone; O - Originate; AD - Disable AA; A/ - Redial.

— Ron Robinson



# BLITZ CON'T

8-BIT

```

8040 IF SC<700 THEN A$="USELESS":GOT
0 8200
8050 IF SC<2000 THEN A$="POOR":GOTO
8200
8060 IF SC<5500 THEN A$="LEARNER":GO
TO 8200
8070 IF SC<8000 THEN A$="NOT BAD!":G
OTO 8200
8080 IF SC<9000 THEN A$="HOT STUFF!"
:GOTO 8200
8090 IF SC<12000 THEN A$="EXCELLENT!
":GOTO 8200
8100 IF SC<19000 THEN A$="ACE!":GOTO
8200
8110 A$="BRILLIANT!!"
8115 IF SP=1 THEN SC=SC-1000
8200 POSITION 6,6:?"YOUR RATING IS:
":A$
8210 POSITION 6,8:?"PRESS START TO
CONTINUE"
8220 IF PEEK(53279)<>6 THEN GOTO 822
0
8230 GOSUB 9000
8240 RETURN
8999 REM INSTRUCTIONS
9000 ? "K":HI=5:SP=1:POKE 82,6:?"FO
R T=53248 TO 53251:POKE T,0:NEXT T
9005 POKE 559,0
9010 ? ",./" ## BLITZ ##
"
9020 ? " [|||||]
"
9030 ? " # FLY YOUR WORLD WAR II"
9040 ? " BOMBER OVER THE DESERTED"
9050 ? " CITY AND TRY TO BOMB ALL"
9060 ? " THE BUILDINGS SO YOU CAN"
9070 ? " LAND SAFELY!"
9080 ? "+ =
"
9081 ? "*" AFTER EACH SUCCESSFUL -
"
9082 ? "*" LANDING YOU ARE AWARDED -
"
9083 ? "*" A BONUS BEFORE GOING ON -
"
9084 ? "*" TO A HARDER CITY -
"
9085 ? "*" -
"
9086 ? "*" <THE FIRE BUTTON BOMBS> -
"
9087 ? "*" PUSH THE JOYSTICK UP TO -
"
9088 ? "*" FIRE OFF YOUR 3 EMERGENCY-
"
9089 ? "*" BOOSTERS FOR MORE HEIGHT!-
"
9090 ? "*= SPEED :FAST: <OPTION> -

```

```

"
9100 ? "*" -
"
9110 ? "*" TONERS :LOW : <SELECT> -
"
9120 ? "*" + +
"
9130 ? "*" PRESS START TO PLAY *-
"
9131 ? "*" = =-
"
9140 ? "*" * * * * * * * * * * * * * * * *
";
9145 POKE 559,46
9150 CON=PEEK(53279)
9155 IF CON=6 THEN RETURN
9160 IF CON=5 AND HI=5 THEN HI=10:PO
SITION 17,19:?"HIGH":GOTO 9210
9170 IF CON=5 AND HI=10 THEN HI=5:PO
SITION 17,19:?"LOW":GOTO 9210
9180 IF CON=3 AND SP=1 THEN SP=0.5:P
OSITION 17,17:?"SLOW":GOTO 9210
9190 IF CON=3 AND SP=0.5 THEN SP=1:P
OSITION 17,17:?"FAST":GOTO 9210
9200 IF CON=6 THEN RETURN
9210 FOR G=1 TO 10:NEXT G:GOTO 9150
9999 REM CRAAAASH!!
10000 POSITION X,Y:?"#6;" ,./":SOUND
0,0,0,0:BOMB=0
10010 GOSUB 11500
10012 FOR G=1 TO 50:NEXT G
10015 SETCOLOR 2,9,8
10016 IF SC<HISC THEN HISC=SC
10020 GOSUB 8000:SC=0:PP=3
10040 ? "K":GOTO 16
11016 IF SC<HISC THEN HISC=SC
11500 FOR S=15 TO 3 STEP -2
11520 X=(S/3)+0.2
11530 FOR T=32 TO 20 STEP -X:POKE 56
0,T:SOUND 1,100,6,5
11540 SOUND 1,60,8,5
11545 SETCOLOR 2,T,T
11550 NEXT T
11560 FOR T=20 TO 32 STEP X:POKE 560
,T:T:SOUND 1,60,8,5
11565 SETCOLOR 2,T,T
11570 NEXT T
11580 NEXT S
11590 POKE 560,32
11600 FOR T=3 TO 0 STEP -0.1
11610 SOUND 1,T*9,0,T
11630 NEXT T
11700 RETURN
12999 REM EXTRA BOOSTER
13000 FOR D=1 TO 50:NEXT D
13010 PP=PP+1
13020 POSITION 20,0:?"PP
13030 FOR T=15 TO 0 STEP -0.5:SOUND

```

```

1,40,10,T:NEXT T
13040 FOR D=1 TO 50:NEXT D
13050 RETURN
19999 REM DEFINE CHARACTERS
20000 DIM DEF$(26):DEF$="-+*,./#&'e
()1234567890_":DIM A$(30),H$(10)
20010 START=(PEEK(106)-8)*256:RDL=57
344:FOR I=0 TO 512:POKE START+I,PEEK
(RDL+I):NEXT I
20015 RESTORE 20050
20020 FOR I=1 TO 26:POS=START+(ASC(D
EF$(I))-32)*8:FOR J=0 TO 7:READ A:PO
KE POS+J,A:NEXT J:NEXT I
20030 FOR I=32 TO 39:POKE START+I,25
5-PEEK(RDL+I):NEXT I:POKE 756,START/
256:RETURN
20050 DATA 24,60,126,255,255,255,255
,255
20060 DATA 255,255,255,129,129,129,2
55,255
20070 DATA 24,24,60,60,102,126,195,2
55
20080 DATA 137,255,145,255,137,255,1
45,255
20090 DATA 32,40,56,63,63,31,0,0,0,2
4,36,255,51,255,0,0,0,32,32,160,224,
160,32,32
20100 DATA 255,24,60,102,66,126,102,
60
20156 DATA 124,198,192,124,6,198,124
,0
20157 DATA 0,124,198,128,128,128,198
,124
20158 DATA 124,198,130,130,130,198,1
24,0
20159 DATA 0,252,134,134,252,240,156
,134
20160 DATA 252,134,128,240,192,134,2
52,0
20161 DATA 24,56,24,24,24,24,60,0
20162 DATA 124,198,134,12,56,98,254,
0
20163 DATA 124,198,2,30,2,198,124,0
20164 DATA 60,96,192,152,222,124,24,
0
20165 DATA 254,128,128,252,6,198,124
,0
20166 DATA 124,230,192,252,198,198,1
24,0
20167 DATA 126,198,12,24,48,48,48,0
20168 DATA 124,198,198,124,198,198,1
24,0
20169 DATA 124,198,198,126,6,198,124
,0
20170 DATA 124,198,138,146,162,198,1
24,0
20180 DATA 255,0,255,0,0,0,0,0
20190 DATA 0,0,0,60,0,60,0,0

```



# BLITZ CON'T

```

20200 DATA 0,0,170,0,170,0,0,0
23999 REM TITLE PAGE
24000 GRAPHICS 0:SETCOLOR 2,3,6:SETC
OLOR 4,1,8
24005 SETCOLOR 1,0,12
24010 POKE 752,1
24020 POKE 82,3:?:?:?
24030?:?:? " A.S.D Presents..."
24040?:?:?
24050?:?:? "
24060?:?:? "
24070?:?:? "
24080?:?:? "
24090?:?:? "
24100?:?:? "
24110?:?:? "
24120?:?:?:? " AN EXPLOSIVE EXPE
RIENCE!"
24130 COLOR 140:PLOT 0,0:DRAWTO 39,0
:DRAWTO 39,23:DRAWTO 0,23:DRAWTO 0,0
:PLOT 0,0
24250 POSITION 0,19
24260?:?:? --PRESS (START)-
--"
24270 GOSUB 30000
24280 SOUND 1,0,0,0:SOUND 2,0,0,0
24500 RETURN
29999 REM COLOURFLOW!
30000 FOR D=1536 TO 1558:READ M:POKE
D,M:NEXT D
30010 DATA 104,173,11,212,101,20,141
,10
30020 DATA 212,141,24,208,173,31,208
,201,6,240,3,76,1,6,96
30025 SOUND 1,240,10,6:SOUND 2,242,1
0,6
30030 A=USR(1536)
30040 RETURN

```

RENEW NOW  
AND AVOID THE RUSH  
CHECK COVER  
TO SEE IF YOU NEED  
TO RENEW

## 8-BIT

Reprinted from Page 6 of England.

### BOMB ESCAPE by Ron Smith

The object is to escape from a building to safe black square before a time bomb explodes.

THE BUILDING--an outline of a building is given in blue with some doors only one pixel wide. Extra rooms, corridors and, sometimes, doors are seen at and above levels three and five. The blue walls must not be touched at any level.

THE TIMER--is ticking away at the bottom of the screen. It is reset if the black square is reached or if the bomb explodes. If a life is lost then the timer continues from the same position. It is not reset to zero. On each new level the timer's fuse is shortened and is not lengthened again during the game.

THE PLAYER--leaves a red trail which must not be touched. The joystick directly controls the speed of the timer--as long as the player is moving, the timer is slowed down. It is a necessity to keep moving at level 5 and above.

LIVES--the game starts with three lives and a life is lost if a red or blue square is touched or if the bomb explodes. If a life is lost then one level is also lost and the game continues at this easier level but with less time available.

GENERAL--an average score of 55,000 is easily possible reaching level 6, however progression beyond this point is difficult and requires expert joystick control.

The timer is controlled by the variables COUNT and TIME. COUNT is increased on line 600 and each time it passes the value of TIME, the program plots another pixel on the timer. TIME is set to 1000 at the start of the game and reduces on line 40 by steps of 100 to a minimum of 400. If either of these variables are altered, then the timer would be lengthened or shortened. DOOR plots two doors at random and is set on line 300.

```

1 REM *****
*
2 REM **          BOMB  ESCAPE          *
*
3 REM **          BY                      *
*
4 REM **          R.F.SMITH. 1984        *
*
5 REM ** ----- *
*
6 REM ** PAGE 6 MAGAZINE - ENGLAND *
*
7 REM *****
*
10 GOTO 1000
20 COUNT=0:X1=0:Y1=47:TIME=1000:LEVEL=1:LIVES=3
30 GOTO 50
40 COUNT=0:X1=0:TIME=TIME-100:IF TIME<400 THEN TIME=400
50 GRAPHICS 21:POKE 712,30:POKE 708,130:POKE 709,0:POKE 710,40
60 RESTORE
70 X3=INT(RND(0)*8+24):Y3=INT(RND(0)*10+15):X2=INT(RND(0)*7+72):Y2=INT(RND(0)*5+1)
100 COLOR 1:PLOT 0,47:DRAWTO 0,0:DRAWTO 79,0:DRAWTO 79,47:PLOT 0,45:DRAWTO 79,45
110 PLOT 5,19:FOR E=1 TO 21:READ A,B:DRAWTO A,B:NEXT E
120 DATA 5,11,13,11,13,4,45,4,45,7,59,7,59,3,71,3,71,7,77,7,77,35,71,35,71,41,59,41,59,37,45,37,45,41,13,41,13,35
130 DATA 5,35,5,24
140 PLOT 65,24:FOR E=1 TO 15:READ A,B:DRAWTO A,B:NEXT E:GOTO 400
150 DATA 65,31,43,31,43,21,43,24,37,24,37,26,23,26,23,13,37,13,37,15,43,15,43,19,43,9,65,9,65,19
160 COLOR 1:FOR E=1 TO 7:READ A,B,C,D:PLOT A,B:DRAWTO C,D:NEXT E:RETURN
170 DATA 7,19,23,19,35,4,35,11,46,19,63,19,66,19,76,19,8,24,23,24,23,28,23,34,23,35,59,35
180 COLOR 1:FOR E=1 TO 7:READ A,B,C,D:PLOT A,B:DRAWTO C,D:NEXT E
190 DATA 15,35,23,35,43,17,51,17,53,19,53,13,65,19,66,19,66,19,65,11,75,11,37,26,37,33,43,11,43,7
200 FOR E=1 TO 9:READ A,B:PLOT A,B:NEXT E:RETURN
210 DATA 77,19,62,4,62,5,68,4,68,5,65,7,65,8,69,9,69,10
300 DOOR=RND(0):IF DOOR>0.5 THEN COLOR 0:PLOT 38,4:PLOT 65,18:RETURN
350 COLOR 3:PLOT 0,47:DRAWTO X1,Y1:RETURN
400 IF LEVEL>2 THEN GOSUB 160:GOSUB

```



# BOMB CON'T

NOV MEETING

```

300
410 IF LEVEL>4 THEN GOSUB 100:GOSUB
300
420 IF LIVES<3 THEN GOSUB 350
430 COLOR 2:PLOT X2,Y2:COLOR 3:PLOT
X3,Y3
440 POKE 77,0:5=STICK(0)
450 IF 5=15 THEN GOTO 600
460 X3=X3+(5=5 OR 5=6 OR 5=7)-(5=9 O
R 5=10 OR 5=11)
470 Y3=Y3+(5=5 OR 5=9 OR 5=13)-(5=6
OR 5=10 OR 5=14)
480 X=X3:Y=Y3
490 COLOR 3:LOCATE X,Y,Z
500 IF Z=3 OR Z=1 THEN GOTO 700
510 IF Z=2 THEN GOTO 900
520 PLOT X,Y
600 COUNT=COUNT+(LEVEL*5)+50:IF COUN
T>TIME THEN GOSUB 650
610 SOUND 0,160,10,6:SOUND 1,200,10,
6:SOUND 0,0,0,0:SOUND 1,0,0,0:GOTO 4
40
650 COLOR 3:PLOT X1,Y1:SOUND 0,100,1
0,14:X1=X1+2
660 IF X1>79 THEN GOSUB 800
670 COUNT=0:SOUND 0,0,0,0:RETURN
700 FOR P=255 TO 20 STEP -2:POKE 710
,15:POKE 700,P:SOUND 0,10,0,10:POKE
710,40:NEXT P:SOUND 0,0,0,0:GOSUB 84
0
710 SOUND 0,0,0,0:SOUND 1,0,0,0:GOTO
50
800 FOR P=30 TO 200:POKE 710,15:POKE
712,P:SOUND 0,P,0,10:POKE 710,32:NE
XT P:SOUND 0,0,0,0:GOSUB 840
830 COUNT=0:X1=0:GOTO 50
840 LIVES=LIVES-1:IF LIVES=0 THEN GO
TO 1100
850 LEVEL=LEVEL-1:IF LEVEL=0 THEN LE
VEL=1
860 RETURN
900 GRAPHICS 18:POKE 712,31:LEVEL=LE
VEL+1:5C=5C+((LEVEL*5)*500-(X1*5)*2)
910 POSITION 5,4:? #6;"LEVEL ";LEVEL
:POSITION 5,6:? #6;"SCORE ";5C
920 FOR DE=1 TO 5:FOR P=100 TO 60 ST
EP -2:SOUND 0,P,10,10:SOUND 1,P+5,10
,10:NEXT P:NEXT DE
930 SOUND 0,0,0,0:SOUND 1,0,0,0:GOTO

```

```

40
1000 GRAPHICS 18:POKE 712,31:POKE 70
8,48:POKE 709,15
1010 POSITION 6,2:? #6;"B":GOSUB 109
0:POSITION 12,4:? #6;"P":GOSUB 1090:
POSITION 6,4:? #6;"5"
1020 GOSUB 1090:POSITION 12,2:? #6;"
B":GOSUB 1090:POSITION 4,4:? #6;"E":
GOSUB 1090:POSITION 14,4:? #6;"E"
1030 GOSUB 1090:POSITION 10,2:? #6;"
M":GOSUB 1090:POSITION 8,4:? #6;"C":
GOSUB 1090:POSITION 8,2:? #6;"Q"
1040 GOSUB 1090:POSITION 10,4:? #6;"
A":GOSUB 1090
1050 DIM A$(15):A$="BY R.F.SMITH.":F
OR X=1 TO 13:POSITION X+2,7:? #6:A$(
X,X):GOSUB 1090:NEXT X
1055 FOR M=1 TO 500:NEXT M:POSITION
3,7:? #6;"
1060 FOR M=1 TO 300:NEXT M:POSITION
4,7:? #6;"press start"
1070 FOR C=191 TO 15 STEP -16:POKE 7
09,C+16:IF PEEK(53279)=6 THEN SOUND
0,0,0,0:GOTO 20
1080 SOUND 0,C/2+10,10,10:FOR M=1 T
O 50:NEXT M:NEXT C:GOTO 1070
1090 SOUND 0,140,2,10:FOR M=1 TO 50:
NEXT M:SOUND 0,0,0,0:FOR M=1 TO 50:N
EXT M:RETURN
1100 GRAPHICS 18:POKE 712,31:POKE 70
9,15:POKE 708,34:POKE 710,34:POKE 71
1,34:SOUND 0,0,0,0:SOUND 1,0,0,0
1110 IF 5C>HI THEN HI=5C
1120 POSITION 8,0:? #6;"BOMB":POSITI
ON 7,1:? #6;"EXPLODE":POSITION 3,7:?
#6;"SCORE ";5C
1130 POSITION 3,8:? #6;"HI SCORE ";H
I
1140 FOR M=1 TO 500:NEXT M:POSITION
4,4:? #6;"press start"
1150 FOR C=191 TO 15 STEP -16:POKE 7
11,C+8:POKE 710,C-6:POKE 709,C+15:PO
KE 708,C-8
1160 IF PEEK(53279)=6 THEN 1180
1170 SOUND 0,C/2+10,10,10:SOUND 1,C/
2,10,10:FOR M=1 TO 50:NEXT M:NEXT C:
GOTO 1150
1180 SOUND 0,0,0,0:SOUND 1,0,0,0:5C=
0:GOTO 20

```

WED THE 12TH

7:30PM

SOUTH EUGENE

HIGH

PLEASE CHECK

COVER DATE

AND RENEW

IF NOV. OR

BELOW

TURBO BASIC GET IT NOW



## Atari Computer Enthusiasts

The ACE Newsletter is the first Atari user group newsletter to be published anywhere in the world. We began publishing in April, 1981.

A.C.E. is an independent, non-profit and tax exempt computer club and user group with no connection to Atari Corp. We are a group interested in educating members in the use of Atari computers and publishing the latest News, Reviews and Rumors.

**All our articles, reviews and programs come from you, our members.**

Our membership is world wide. Membership fees include the A.C.E. newsletter (10 issues per year) and high access in the A.C.E. Bulletin Board System. Dues are \$14 a year for U.S., and \$24 a year overseas airmail.

**Subscription Dept.:** 3662 Vine Maple Dr., Eugene, OR 97405

**President** — Dick Barkley, 2907 Wingate, Eugene, OR 97405  
(503) 344-5843

**Vice President** — Larry Gold, 1927 McLean Blvd., Eugene, OR 97405  
(503) 686-1490

**8-Bit Librarian** — Chuck & Jody Ross, 2222 Ironwood, Eugene, OR 97401  
(503) 343-5545

**ST Librarian** — Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405  
(503) 484-4746

**Editors** — Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405  
(503) 344-6193

Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405  
(503) 484-4746

Larry Gold, 1927 McLean Blvd., Eugene, OR 97405  
(503) 686-1490

**E.R.A.C.E.** (Education SIG Editor) — Nora Young, 105 Hansen Lane  
Eugene, OR 97404 / (503) 688-1458

Send 50c stamps or coin (\$1 overseas) to the Librarian for updated ACE Library lists.

## A.C.E. Bulletin Board (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a 130 XE upgraded to 320k by Charlie Andrews, 2 double-density disk drives, and a 10-meg hard disk from Supra Corp, an Epson MX80 printer, a 2400 US Robotics Courier modem, running the PD/M Bulletin Board software.

### Best of ACE books

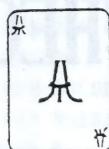
Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct. 1981 to June, 1982.

Volume 2 contains July 1982 through June 1983.

Only \$12 each (\$2 extra for airmail). Available only from George Suetsugu, 45-602 Apuapu St., Kanoeha, HI 96744.

### SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including the ACE Newsletter. Volume 1 covers April, 1981 through June, 1983. Volume 2 covers July, 1983 through December, 1985. Only \$6 per printed copy or \$11 per disk copy for ACE members. Available from: Jim Carr, 2660 S.W. DeArmond, Corvallis, OR 97333.



ATARI  
COMPUTER  
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

**BULK RATE**  
**U.S. POSTAGE**  
**PAID**  
**PERMIT NO. 472**  
**EUGENE, OREGON**

**RENEW  
NOW**

**==>**

**IF DATE EARLIER**

**THAN NOV 86**